

## Treball de fi de grau

Grau en enginyeria electrònica industrial i automàtica

Actualització Robot Braccio, visualització d'angle

**Autor :** Hiroshi Morinaka Arias

**Director:** Antoni Escobet Canal

**Convocatòria** octubre de 2019

## **Resum**

El present projecte de final de grau té com a objectiu principal millorar les prestacions del braç robòtic que utilitzen a la universitat a través d'un visualitzador d'angles.

Aquesta actualització permetrà als alumnes de la universitat, facilitar la programació del robot i sigui una eina més precisa per aconseguir els nostres objectius, en les tasques que volem desenvolupar.

La millora serà utilitzar servomotors amb feedback (Retroalimentació), que farà que el robot braccio pugui llegir l'angle que ha fet i poder així, enviar-ho directament al monitor sèrie del programa d'Arduino, on es podrà veure els angles que estan fent cada un d'ells.

## **Abstract**

The main goal of this final degree project is to improve the robotic arm that they use in the university with an angle viewer.

This update allows university students to facilitate the programming of the robot and be a more accurate tool to achieve our goals to the tasks they want to develop.

The upgrade will be to use servomotor with feedback, which will make the robot braccio read the angle he has made and send it directly to the monitor series of the Arduino program. where you can see the angles that each servomotor is doing.

# Índex

1. Introducció .....	1
1.2 Origen del treball.....	1
1.3 Motivació.....	1
1.4 Requeriments previs .....	2
1.5 Objectius del treball .....	3
1.6 Abast de treball .....	3
2. Antecedent.....	3
2.1 Introducció a la Robòtica.....	4
2.2 Tipus de robots.....	4
2.3 Programació de robot .....	13
2.3.1 Programació Híbrida.....	14
2.4 Estudi de mercat .....	15
2.5 TinkerkítBraccio Robot .....	19
2.5.1 Característiques.....	20
3. Disseny electrònic .....	23
3.1 Selecció de components.....	23
3.1.1 Microcontrolador .....	23
3.1.2 Servomotor Feedback .....	25
3.1.3 Shield Robot braccio.....	30
4. Programació del robot braccio.....	33
4.1 Estructura d'un sketch .....	33
4.2 Desenvolupament de l'algoritme.....	35
4.2.1 Declaració de pins .....	35
4.2.2 Calibratge de valors.....	35
4.2.3 Inicialització del servo .....	36
4.2.4 Bucle de funcionament .....	36
5. Conclusions.....	38
6. Estudi Econòmic .....	39
7. Bibliografia .....	40
8. Annex A: Codi implementat En Arduino .....	41

# Índex d'il·lustracions

Il·lustració 1 Tinkercat Braccio Robot .....	2
Il·lustració 2 Robot cartesians .....	5
Il·lustració 3 Robot Dual-arm .....	6
Il·lustració 4 Tipus de robot industrials, classificats per la seva estructura, i dels quals es detalla l'àrea de treball i es mostra un exemple real.....	7
Il·lustració 5 iRobot Romba .....	8
Il·lustració 6 Robot neteja fons .....	9
Il·lustració 7 kit de muntatge .....	9
Il·lustració 8 exosquelet .....	10
Il·lustració 9 DaVinci "Surgery" .....	11
Il·lustració 10 Exemple de Programació híbrida llenguatge en C.....	15
Il·lustració 11 control amb el programa RIOS S370120 .....	16
Il·lustració 12 Braç Robòtic.....	17
Il·lustració 13 projecte de muntatge i control d'un robot fet de materials reciclats.....	17
Il·lustració 14 Exemple de projecte de muntatge i control d'un robot.....	18
Il·lustració 15 Exemple de projecte usant el kit complet del robot MeArm .....	18
Il·lustració 16 Contingut de la caixa del Tinkercat Braccio Robot seguint el manual de utilització .....	19
Il·lustració 17 Contingut de la caixa del Tinkercat Braccio Robot.....	20
Il·lustració 18 Contingut de la caixa del Tinkercat Braccio Robot.....	20
Il·lustració 19 Servomotor SR311 i Sr431.....	22
Il·lustració 20 Estructura de la placa de desenvolupament d'Arduino Uno Rev3 .....	25
Il·lustració 21 Sistema bucle obert.....	26
Il·lustració 22 Sistema de circuit tancat .....	26
Il·lustració 23 Funcionament servomotor Standard .....	27
Il·lustració 24 Funcionament servomotor feedback .....	27
Il·lustració 25 Connexió servomotor feedback .....	29
Il·lustració 26 Analog Feedback Servo.....	29
Il·lustració 27 Analog Feedback Micro servo .....	30
Il·lustració 28 Shield.....	30
Il·lustració 29 Connexió shield amb l'Arduino Uno Rev3.....	31
Il·lustració 30 Esquema intern de la shield v1 .....	32
Il·lustració 31 Implementació de les funcions en Arduino IDE .....	34
Il·lustració 32 Visualització d'angles al monitor sèrie amb un servomotor amb feedback .....	36
Il·lustració 33 Visualització d'angles al monitor sèrie amb més d'un servomotor amb feedback .....	37
Il·lustració 34 Visualització angles al monitor amb més d'un servomotor amb feedback.....	37

## Índex Taula

Taula 1 classificacions dels robots de servei per l'ús domèstic i personal.....	11
Taula 2 classificacions dels robots de servei destinats a l'ús Professional .....	12
Taula 3 característiques tècniques del robot Braccio .....	21
Taula 4 Característiques tècniques de servomotor SR311 i Sr431-Dual Output Servo .....	22
Taula 5 Característiques de servomotor feedback .....	28
Taula 6 Estudi econòmic.....	39

## 1. Introducció

Aquest treball de final de grau (TFG) està orientada a la millora del funcionament del braç robòtic TinkerkitBraccio Robot (Robot Braccio en endavant) amb un visualitzador d'angle. Per tal de fer possible que la utilització de Robot Braccio sigui més eficient i més fàcil d'utilitzar.

El fet de poder visualitzar els angles de cada servo del braç robot, facilitaria la programació del robot Braccio i una millora d'eficiència en el desenvolupament de les tasques que volem desenvolupar.

### 1.2 Origen del treball

Aquest treball s'origina en la intenció de millor el robot Braccio, ja que té unes limitacions que a l'hora de fer algun projecte o treball de classe dificulta la execució d'aquestes tasques

Els motius perquè hem decidit fer aquest projecte són els següents:

1. Problemes en veure si els angles estan ben posicionats
2. Monitoritzar independentment cada un dels servos del robot per veure si estan fent els angles que hem programat.
3. Tenir un registre de moviments del robot per poder veure si cada cicle de moviments que fa, és igual als moviments programats.

### 1.3 Motivació

La inclinació de l'autor cap a l'automatització i la robòtica cultivada al llarg dels anys d'especialització ha originat l'interès per aquest TFG que implica realitzar una millora d'un producte comercial de propòsit educatiu i d'oci com és el TinkerkitBraccio Robot, de manera que els nous alumnes de l'Escola Politècnica superior d'Enginyeria de Manresa puguin gaudir d'una experiència en la robòtica eficient en la programació i sense limitacions.



**Il·lustració 1 Tinkerkit Braccio Robot**

Un altre de les motivacions per realitzar el treball és també la predisposició i ganes d'aprendre nous llenguatges i, entorns de programació i muntatge en àmbits de la robòtica.

## **1.4 Requeriments previs**

Ja que la programació és important en aquest treball, tenir coneixements previs sobre el llenguatge que utilitza Arduino facilitaria en l'àmbit de la programació, però treballar en Arduino es pot aprendre de forma autònoma, ja que es una programació senzilla i poden trobar molta informació de programació en Internet o manuals de la pròpia marca.

D'altra banda, els coneixements sobre electrònica son també rellevants, ja que es poden solucionar possibles problemes de maquinària o components electrònics, en cas que sigui necessari. A més dels coneixements tècnics previs, es treballaran i reforçaran les habilitats de planificació del treball, la capacitat de dividir els problemes en petits parts per així facilitar la seva resolució i l'ús solvent dels recursos de la informació.

En resum, es podria dir que tots els coneixements, adquirits durant el grau en enginyeria electrònica industrial i automàtica han estat rellevant en la realització del present treball.



## 1.5 Objectius del treball

El principal objectiu del treball és millorar el robot que utilitzem a la universitat, per poder desenvolupar projectes més complexos i poder entendre millor la robòtica. Aquesta millora es farà amb un servomotor amb feedback, per poder veure els angles que treballa cada servomotor que conformen el robot , i poder monitoritzar el correcte funcionament de cada un d'ells

## 1.6 Abast de treball

Aquest treball és d'abast variable atès que es pot millorar cada vegada més amb diferents sensors que podem incorporar al robot. En el subapartat anterior s'ha presentat l'objectiu principal, és a dir, aquell objectiu mínim que s'ha de complir i consta de:

1. Programar els servomotors amb feedback per veure els angles al monitor sèrie del programa Arduino.
2. Integració dels nous servos al Robot Braccio

## 2. Antecedent

Abans d'abordar el desenvolupament d'aquest treball, fan falta un conjunt de conceptes generals sobre robots, els quals es descriu en aquesta part.

## 2.1 Introducció a la Robòtica

La robòtica va lligada a la construcció "d'artefactes" que tractaven de materialitzar el disseny humà de crear éssers a la seva semblança i que al mateix temps ho descarreguessin de treballs tediosos. (Leonardo Torres Quevedo, 1914) en els seus Assajos sobre automàtica va encunyar el terme "automàtica" en relació amb la teoria de l'automatització de tasques tradicionalment associades.

(Karel Čapek, 1920), un escriptor txec, va encunyar el terme "robot" en la seva obra dramàtica Rossum's Universal Robots / RUR, a partir de la paraula txeca "robota", que significa servitud o treball forçat. (Asimov, 1950) en la seva obra tres lleis de la robòtica va encunyat el terme "robòtica", definint a la ciència que estudia als robots.

Paral·lelament en el desenvolupament d'altres camps com l'electrònica i la informàtica, les característiques dels robots han millorat, de manera que el seu ús ha començat a sortir en l'àmbit industrial, com es descriurà a continuació en la classificació dels robots.

## 2.2 Tipus de robots

Segon la normativa ISO (Organització Internacional per l'estandardització) 8373, preparada pel comitè tècnic ISO / TC 184, Sistemes d'automatització i integració, Subcomitè SC 2, Robots i dispositius, revisada per última vegada en 2012.

Els robots es classifiquen en dos grans grups:

**Robots industrials:** Segons la norma ISO 8373, un robot industrial és un manipulador multipropòsits, reprogramable i controlat automàticament, de tres o més eixos, que poden ser fixos o mòbils per un ús en aplicació d'automatització industrial i que segons la seva estructura mecànica, es classifica en:

- Robot lineals :
- **robots cartesianes:** Posseeix tres articulacions prismàtiques que, en ser perpendiculars, coincideixen amb el sistema de coordenades cartesianes.



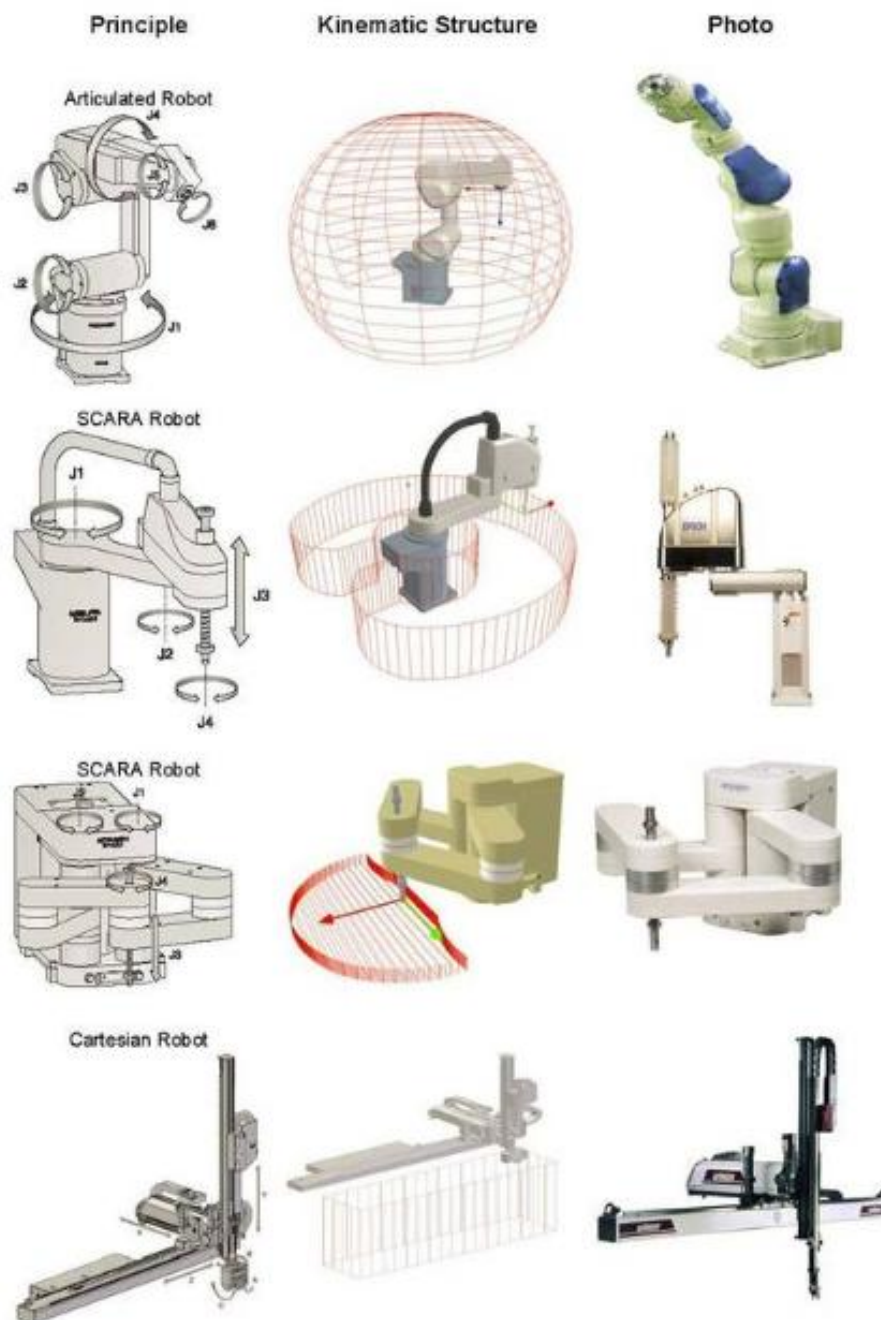
Il·lustració 2 Robot cartesianes

- **Robots "Gantry":** Consisteix en robot manipulador muntats en un sistema suspès que permet el seu moviment a través d'un pla horitzontal.
- **Robot SCARA:** Es tracta d'un robot amb dos eixos de rotació i un lineal, paral·lels entre ells. Poden tenir un eix addicional per orientar l'extrem.
- **Robot Articulats:** Consta de tres eixos rotacionals, un perpendicular a la base i els altres dos paral·lels entre si i perpendiculars al primer eix.
- **Robots paral·lels o Delta:** Són robots formats per articulacions prismàtics o rotatoris concurrents(cadena cinemàtica tancada).
- **Robots cilíndrics:** Estan formats per dos eixos prismàtics perpendiculars i un eix de rotació vertical sobre la base que formen un sistema de coordenades cilíndriques.
- **Altres:**
  - Robot "Dual-arm":** Estructura formada per dos braços articulats.



**Il·lustració 3 Robot Dual-arm**

En la il·lustració 4 podem veure els tipus de robots descrits anteriorment .



Il·lustració 4 Tipus de robot industrials, classificats per la seva estructura, i dels quals es detalla l'àrea de treball i es mostra un exemple real.

**Robot de servei** : Són aquells Robots, que l'aplicació es troba fora de l'entorn industrial i dóna algun tipus de servei als humans. Segons la seva definició en la norma ISO, requereix cert grau d'autonomia, anant des d'una autonomia parcial fins a una total, és a dir, amb o sense quasi interacció huma-robot, ja que, es denomina com robot amb total autonomia. Es declara que han de tenir un cert grau d'interacció amb un operador. Dintre d'aquest grup, es poden diferenciar en dos tipus:

- **Robots de servei per ús personal:** L'usuari d'aquest tipus de robot no requereix coneixements tècnics previs sobre el robot per poder utilitzar. La federació internacional de Robòtica (IRF) distingeix els següents tipus:
- **Robots de tasques domèstics:** Amb l'objectiu de facilitar i agilitzar les tasques domèstiques, que dintre d'una societat frenètica com l'actual representa tot un inconvenient, aquests robots poden ser, per exemple, tallagespa, netejafons.



Il·lustració 5 iRobotRoomba



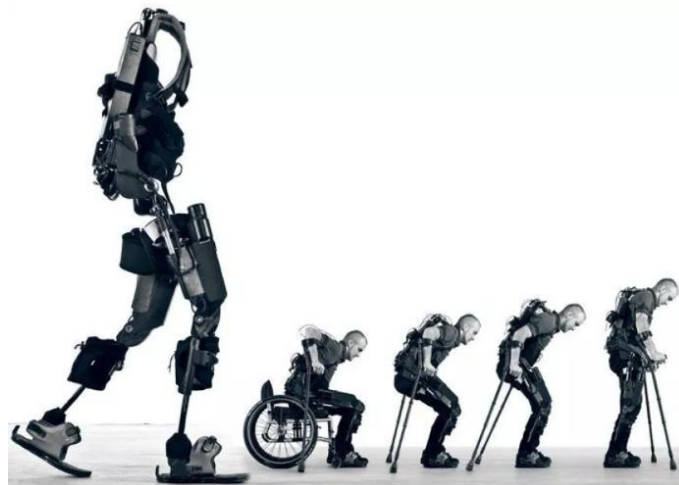
Il·lustració 6 Robot neteja fons

- **Robot d'entreteniment:** no compleix una tasca específica. Poden tractar-se de joguines, robots educatius, kits de muntatge de robot, "robot rides".



Il·lustració 7 kit de muntatge

- **Robot d'assistència als discapacitats:** A través de la contínua millora en les característiques dels robots, actualment, es desenvolupa aplicacions que ajuden en el dia a dia de persones amb algun tipus de minusvalidesa. Algun exemple d'aquest tipus de robot són les cadires de rodes robotitzats, robot d'assistència personal capaç d'interactuar amb l'usuari, plataformes elevadores, salva escales o exosquelets.



Il·lustració 8 exosquelet

- Transport personal-AGV(AutomatedGuide Vehicles)per persones.
- seguretat i vigilància domestica.
- **Robot de servei d'ús Professional:** En aquest cas al contrari que l'anterior, és necessari que l'usuari u operari posseeix alguna formació prèvia, de manera que sigui capaç d'iniciar el robot o el sistema robòtic, monitoritzar i parar-lo. Aquest tipus de robot es poden trobar en diversos àmbits amb l'agricultura, la ramaderia, la mineria, la neteja professional o la medica.





Il·lustració 9 DaVinci "Surgery"

Aquesta classificació es poden veure de forma més detallada en la taula 2 i 3

Taula 1 classificacions dels robots de servei per l'ús domèstic i personal.

Section I	Types of robots: Service robots for personal/domestic use
1-6	Robots for domestic tasks
1	Robot companions/assistants/humanoids
2	Vacuuming, floor cleaning
3	Lawn-mowing
4	Pool cleaning
5	Window cleaning
6	Others
7-10	Entertainment robots
7	Toy/hobby robots
8	Multimedia/remote presence
9	Education and research
10	Others
11-13	Elderly and handicap assistance
11	Robotized wheelchairs
12	Personal aids and assistive devices
13	Other assistance functions
14	Personal transportation (AGV for persons)
15	Home security & surveillance
16	Other Personal / domestic robots

Taula 2 classificacions dels robots de servei destinats a l'ús Professional

Section II	Types of robots: Service robots for professional use
17-23	Field robotics
17	Agriculture
18	Milking robots
19	Other robots for livestock farming
20	Forestry and silviculture
21	Mining robots
22	Space robots
23	Other field robotics
24-28	Professional cleaning
24	Floor cleaning
25	Window and wall cleaning (incl. wall climbing robots)
26	Tank, tube and pipe cleaning
27	Hull cleaning (aircraft vehicles etc.)
28	Other cleaning tasks
29-31	Inspection and maintenance systems
29	Facilities, plants
30	Tank, tubes, pipes and sewers
31	Other inspection and maintenance systems
32-35	Construction and demolition
32	Nuclear demolition & dismantling
33	Building construction
34	Robots for heavy/civil construction
35	Other construction and demolition systems
36-39	Logistic systems
36	Automated guided (AGV) vehicles manufacturing environments
37	AGVs non-manufacturing environments (indoor)
38	Cargo handling, outdoor logistics
39	Other logistic systems
40-43	Medical robotics
40	Diagnostic systems
41	Robot assisted surgery or therapy
42	Rehabilitation systems
43	Other medical robots
44-46	Rescue & security applications
44	Fire and disaster fighting robots
45	Surveillance / security robots
46	Other rescue and security robots
47-50	Defense applications
47	Demining robots
48	Unmanned aerial vehicles
49	Unmanned ground based vehicles
50	Unmanned underwater vehicles
51	Other defense applications
52	Underwater systems (civil / general use)
53	Powered Human Exoskeletons
54	Unmanned aerial vehicles (general use)
55	Mobile Platforms in general use

56-60	Underwater systems (civil / general use)
56	Hotel & restaurant robots
57	Mobile guidance, information robots
58	Robots in marketing
59	Robotjoy rides
60	Others (i.e. library robots)
61	Other professional service robots not specified above

## 2.3 Programació de robot

Programar, en aquest context, vol dir descriure un conjunt d'accions consecutives que el robot realitza per tal de completar una tasca determinada. Aquestes accions passen pel moviment de l'element terminal a un punt prèviament definit, descrivint o no una trajectòria, amb l'objectiu de manipular o processar un objecte, amb l'ajuda o sense elements externs que controlen el robot i interactuar amb el seu entorn

En la programació es distingeix tres tipus o mètodes :

- Programació per guiatge o " on-line"
- Programació textual o off-line
- Programació Híbrida

Però ens enfocarem en la programació Híbrida, Perquè és la programació que utilitzarem en el nostre treball.

### 2.3.1 Programació Híbrida

La programació híbrida s'utilitza en casos on el codi en ensamblador dificulta l'estructura del programa. La programació híbrida proporciona un mecanisme per mitjà del qual podem aprofitar els avantatges del llenguatge ensamblador i els llenguatges d'alt nivell, tot això amb la finalitat d'escriure programes més ràpids i eficients.

Treballar amb un llenguatge d'alt nivell, de vegades ens troben amb el problema que necessitem que faci determinada funció o treball, però desafortunadament aquesta només existeix en un altre llenguatge que no és el que necessitem utilitzar, o simplement, no troben aquesta funció en cap llenguatge d'alt nivell

Avantatge de la programació Híbrida

- Millorar l'escalabilitat
- Quan moltes tasques produeixen desequilibri
- Reducció de temps de desenvolupament de codi
- Quan el nombre de processos MPI és fix
- En cas de mescla de paral·lelisme funcional i de dades

En aquest moment el llenguatge ensamblador constitueix una eina no només eficaç, sinó simple per produir una solució per al compilador del nostre llenguatge preferit. Potser el major problema amb què ens enfrontem sigui el de com connectar els dos programes (el d'alt i el de baix nivell) i com passar variables d'un programa a l'altre. Per aconseguir el nostre objectiu s'utilitzen pseudo-operadors, es a dir, instruccions que apareixen en codi font del ensamblador però que no generi cap instrucció de maquinària, però proporcionen directives perquè l'ensamblador pugui operar amb dades, ramificacions condicionals, generació de llistat i amb macros durant el procés d'acoblament. En la il·lustració 10 veurem un exemple de programació híbrida.

```
255 function updatePhotoDescription() {  
256   if (descriptions.length > (page * 9) + (currentImage.substring(0, 3)))  
257     document.getElementById('bigimageDesc').innerHTML = descriptions[  
258   ]  
259 }  
260  
261 function updateAllImages() {  
262   var i = 1;  
263   while (i < 10) {  
264     var elementId = 'foto' + i;  
265     var elementIdBig = 'bigimage' + i;  
266     if (page * 9 + i - 1 < photos.length) {  
267       document.getElementById( elementId ).src = 'images/  
268       document.getElementById( elementIdBig ).src = 'image  
269     } else {  
270       document.getElementById( elementId ).src = '';
```

Il·lustració 10 Exemple de Programació híbrida llenguatge en C

## 2.4 Estudi de mercat

Tenint en compte la forta presència dels robots en el mercat actual per les seves diverses configuracions i la diversitat que presenta cadascuna d'elles, l'objectiu del present treball és millorar i afegir eines noves per millorar el producte que es presentaran al següents subapartats.

En l'actualitat la realització de projectes que incloguin el muntatge d'un braç robòtic i la programació dels moviments d'aquest producte és, molt extensa. Tot i aquests, l'ús que se'ls dona està orientada únicament al moviment de cada articulació i a realitzar aquests moviments, prèviament definits, de manera consecutiva. Aquells projectes que van més enllà d'aquest objecte, solen ser productes acabats que es venen al mercat com kits de muntatge amb programari inclòs, tant per a microcontroladors o per a aplicacions des d'on es controla, que pot ser un comandament, un dispositiu mòbil, tablet o una aplicació d'ordinador. En aquest treball es pretén afegir eines noves al braç robòtic, per desenvolupar projectes més difícils i més variats de tasques a resoldre .

A la pàgina web d'Arduino dedicada als projectes amb plaques de desenvolupament d'Arduino ([Http://playground.arduino.cc/](http://playground.arduino.cc/) o <https://create.arduino.cc/projecthub>), es pot trobar projectes sobre la construcció de braços robòtics, dels quals, aquells controlats des d'un altre dispositiu, solen utilitzar aplicacions pre fetes, és a dir, aquelles que serveixen per moure el robot i que ja conté un protocol de comunicació que s'ha d'adaptar en el projecte o que són fetes pel fabricant del robot. En el cas de programar el microcontrolador, molts dels projectes ho fan a través d'aplicacions o entorns de programació més senzilles (programació en visual) per a aquelles persones amb poques nocions de programació, com ara snap4Arduino o scratch for Arduino (S4A), les quals són versions de snap! i scratch, respectivament, orientades a la programació dels microcontroladors de les plaques de desenvolupaments d'Arduino.

A continuació, es procedeix a explicar alguns exemples de projectes que utilitzen llenguatge snap4arduino o scratch for arduino:

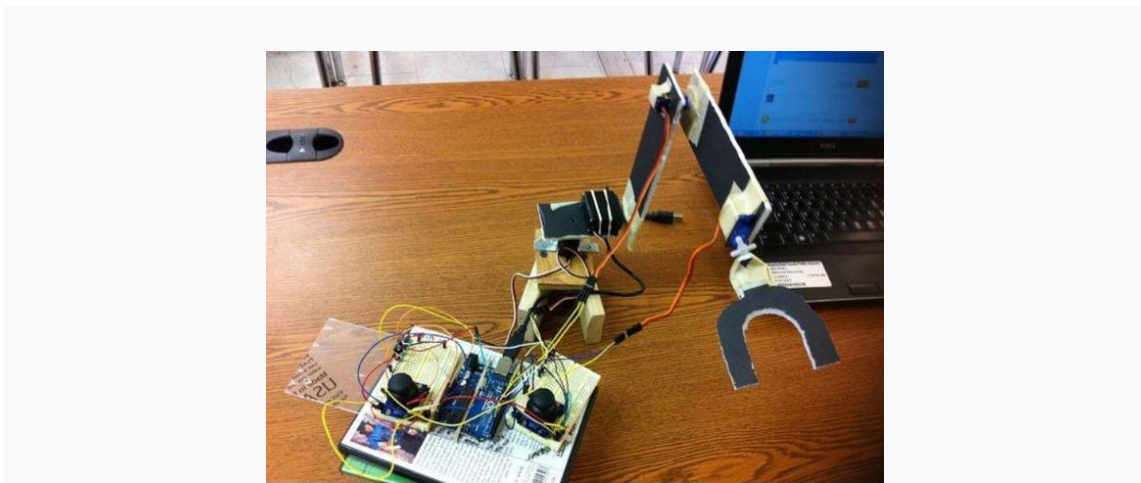


Il·lustració 11 control amb el programa RIOS S370120



**Il·lustració 12 Braç Robòtic**

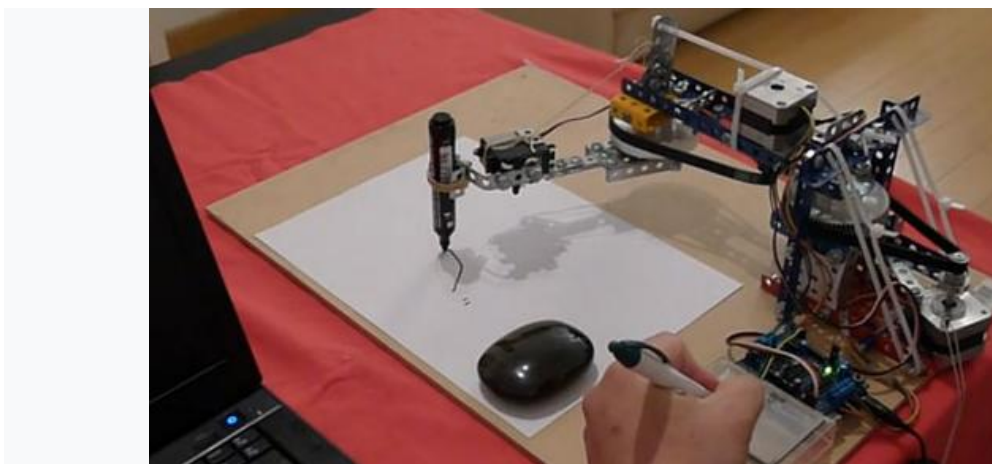
En la il·lustració 11 veiem el robot AL5B i en la il·lustració 10 el programar RIOS S370120. El programa inclou funcions avançades per a la realització de tota mena de moviments amb el braç robot incloent compensació de gravetat i pes. També Permet utilitzar diferents eines per controlar el braç robot com per exemple un joystick.



**Il·lustració 13 projecte de muntatge i control d'un robot fet de materials reciclats**

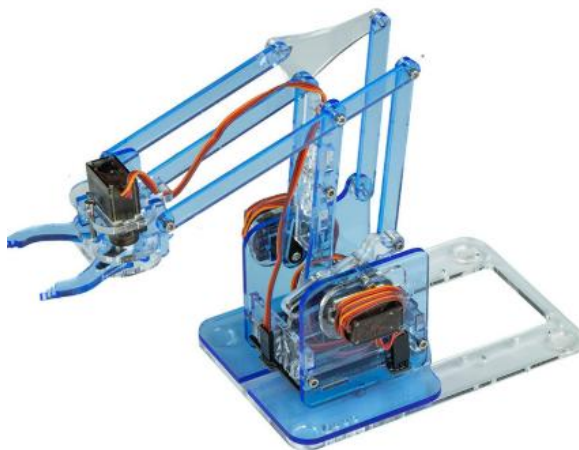
En la Il·lustració 12 podem veure la construcció d'un braç robòtic a partir de materials reciclats, una placa de desenvolupament Arduino / Genuí Un i alguns Shields d'Arduino que permeten el control del robot amb un joystick.





**Il·lustració 14 Exemple de projecte de muntatge i control d'un robot**

En la il·lustració 13 podem veure un braç robot desenvolupat per Acorv que és capaç de dibuixar sobre un full de paper utilitzant un petit retolador, però això no és tot, ja que l'interessant és que ho fa a partir d'una pantalla tàctil resistiva.



**Il·lustració 15 Exemple de projecte usant el kit complet del robot MeArm**

En la Il·lustració 14 es pot observar un projecte orientat al moviment del robot MeArm a través d'un Smartphone, utilitzant una placa de desenvolupament Arduino / Genuí i Un l'Shield d'Android 1Shield, el qual compta amb una aplicació mòbil que permet utilitzar els components que incorpora qualsevol Smartphone actual. En aquest cas, es fa ús del giroscopi i el sensor de proximitat d'un mòbil per moure el robot.

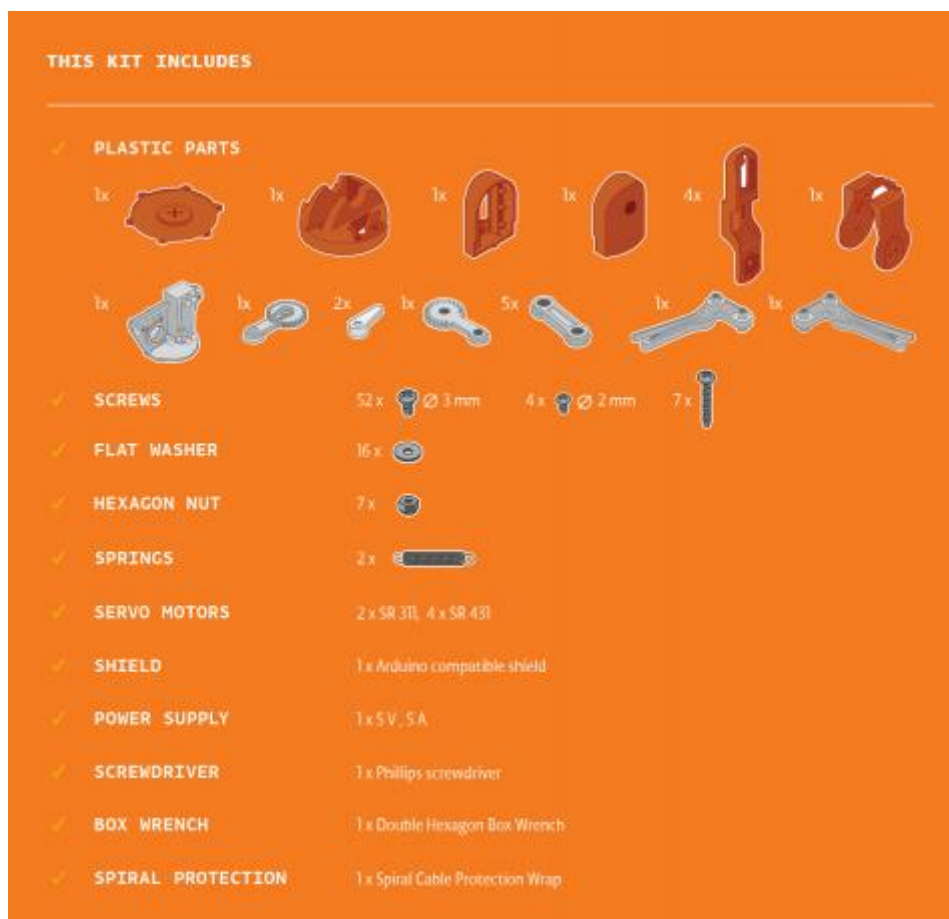
Altres exemples es poden trobar en blocs o en la plataforma de Youtube en forma de tutorials o presentació del resultat del projecte realitzat.



Com s'ha pogut observar, hi ha tota una gamma de kits semblants al TinkerkitBraccio Robot, amb diferents característiques i de diferents preus. Cal afegir que una alternativa als kits, que pot ser un producte més barata, és l'adquisició de robots d'oci, els quals, amb un parell de coneixements d'electrònica, es poden personalitzar per a ser programables.

## 2.5 TinkerkitBraccio Robot

En aquest treball s'utilitzarà el Tinkerkit Robot d'Arduino, un kit de muntatge d'un braç robòtic fet de peces de plàstic i 6 servomotors, els quals són controlats per un Shield dissenyat per superposar en una placa de desenvolupament de l'Arduino. El preu ronda actualment uns 199 euros amb missatgeria.



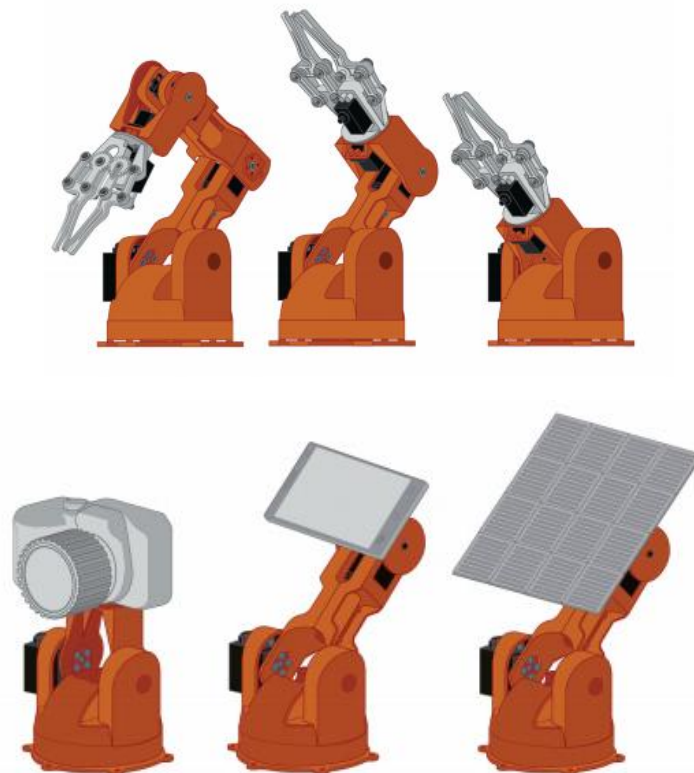
Il·lustració 16 Contingut de la caixa del TinkerkitBraccio Robot seguint el manual de utilització



Il·lustració 17 Contingut de la caixa del Tinkerkit Braccio Robot

### 2.5.1 Característiques

Una de les característiques més notables és que la seva estructura no és única, per tant és possible adaptar-la a l'aplicació diferents com es veu en la il·lustració 18



Il·lustració 18 Contingut de la caixa del Tinkerkit Braccio Robot

Respecte a les especificacions més tècniques, es resumeix en la taula 4, la qual mostra les característiques més generals i, seguidament, en la taula 5 amb els detalls dels servomotors que te el kit Tinkerkit Robot d'Arduino.

**Taula 3 característiques tècniques del robot Braccio**

Característiques	Detalls
<b>Pes</b>	0,792 Kg
<b>Distancia de funcionament</b>	80 cm
<b>Màxima Altura</b>	52 cm
<b>Amplada de la base</b>	14 cm
<b>Amplada de la pinça</b>	9cm
<b>Longitud del cable</b>	40 cm
<b>Capacitat de carga</b>	0,15 kg/32cm
<b>Capacitat de carga en configuració mínima</b>	0,4
<b>Font d'alimentació</b>	5V / 5A
<b>Consum del Shield</b>	0,02W
<b>Corrent Màxima de la Shield</b>	De M1 a M4 1,1A De M5 a M6 0,75A
<b>Servomotors</b>	2x SR 311+4x SR431
<b>Mesures de la Shield PCB</b>	6,858x5,334 cm

**Taula 4 Característiques tècniques de servomotor SR311 i Sr431-Dual Output Servo**

Servomotor SR 311		Servomotor SR 431-Dual	
Característiques	Detall	Característiques	Detall
Senyal de control	PWM Analògica	Senyal de control	PWM Analògica
Esforç de torsió	4,8V:3,1kg.cm 6v :3,8 kg.cm	Esforç de torsió	4,8V:12,2kg.cm 6v :14,5 kg.cm
Velocitat	4,8V:0,14s/60º 6v :0,12s/60º	Velocitat	4,8V:0,20s/60º 6v :0,18s/60º
Rang de rotació	180º	Rang de rotació	180º
Dimensions	31,13 x16,5x28,6 mm	Dimensions	42,0x20,5x39,5
pes	0,027kg	pes	0,062

**Il·lustració 19 Servomotor SR311 i Sr431**

### 3. Disseny electrònic

En aquesta secció es presenta els materials utilitzats, junt amb els motius que han portat a terme a la selecció.

#### 3.1 Selecció de components

El kit de muntatge utilitzat no inclou el microprocessador que ha de gestionar el moviment del robot, per tant hem elegit un que s'ajusta a l'aplicació que es donarà aquest treball

Per altra part necessiten un servomotor amb feedback, ja que els servomotors convencionals sense feedback no podem recollir les dades necessàries per poder trobar els angles de cada servo que nosaltres volem.

##### 3.1.1 Microcontrolador

Un microcontrolador és un sistema electrònic digital programable compost d'una unitat central de processament o CPU, una memòria i unitat d'entrada i sortida.

Inicialment s'han plantejat com a opcions les plaques de desenvolupament de les marques Arduino i Raspberry, ja que la utilització d'aquest en les construccions de prototips és molt extensa, per la flexibilitat que ofereix per ampliar i modificar el muntatges, la quantitat d'informació que pot emmagatzemar i es pot compilar i executar des de qualsevol entorn de programació. Per part d'Arduino, els seus avantatges són la facilitat que presenta de crear aplicacions amb les seves llibreries pròpies, sense utilitzar unes rigoroses nocions de programació i en cas de la Raspberry, les característiques tècniques superiors a la de l'arduino que ofereix de baix cost.

Dels dos microcontroladors hem elegit l'Arduino perquè el kit que utilitzen és de la marca arduino i teníem més nocions en l'entorn del programa. El model de microcontrolador d'Arduino que hem elegit és l'Arduino uno Rev3. Un altre motiu és que la Shield que alimenta els servomotors del Robot Braccio està fet per utilitzar amb una placa de desenvolupament d'Arduino, encara que no és obligatori, ja que es pot adaptar per treballar amb raspberry.

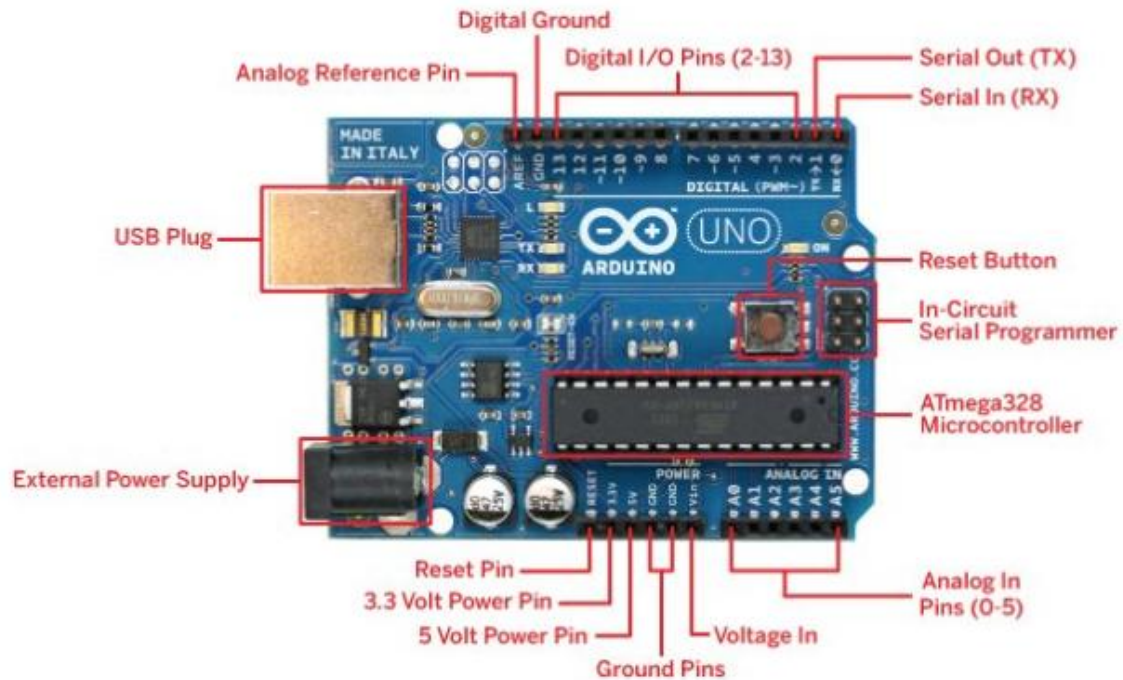
El microcontrolador ATmega328p de 8 bits, en el que està basat la placa de desenvolupament d'Arduino uno, presenta les següents característiques:

- Alt rendiment
- Baix consum amb 5 maneres d'estalvi de consum per softwar
- 32 KB de memòria de programa, tipus Flash, de els quals 0,5 KB son destinats al *bootloader*
- 2KB de memòria SRA
- 1KB de memòria EEPROM
- Velocitat de la CPU de 20 MPIS (milions de instruccions per segons)
- 23 pins de entrada/sortida
- 2 temporitzador/controlador de 8 bits y 1 de 16 bits habilitats amb mode de comparació
- Interrupcions intern i externs
- Perifèrics de comunicacions digitals :USART(" Universal Synchronous and Asynchronous Receiver-Transmitter") programable, 2 ports SPI("Serial Peripheral Interface") i 1 bus I2C
- Un conversor analògic digital de 10 bits y 6 canals
- Altres perifèrics :6 PWM("Phase Width Modulation"), 1 CCP ("Capture/Compare/PWM"), 1 "Input Capture"
- Rang temperatura de funcionament de -40 a 85 °C
- Rang de voltatge de funcionament de 1,8 a 5,5 V

Per altre part, les característiques tècniques de la placa de desenvolupament son la següent :

- Voltatge de funcionament de 5 V
- Voltatge d'entrada recomanat de 7 a 12 V
- Voltatge d'entrada límits de 6V i 20 V
- 14 pins d'entrada/sortida digital, 6 dels quals proporcionen una sortida PWM
- 6 pins d'entrada analògica
- Corrent continu per els pins d'entrada/sortida de 20 mA
- Corrent continu per els Pins de 3,3 V de 5mA
- Velocitat de rellotge de 16 MHz
- Proporcionen un LED muntat en la placa , que es pot controlar des de el pin digital 13
- Dimensions 6,86x5,34 cm
- Pes 25 g

En la il·lustració 19, es presenta l'estructura de la placa de desenvolupament Arduino Uno Rev3.



Il·lustració 20 Estructura de la placa de desenvolupament d'Arduinouno Rev3

### 3.1.2 Servomotor Feedback

Servo és un terme general per a un sistema de control de bucle tancat que utilitza retroalimentació negativa.

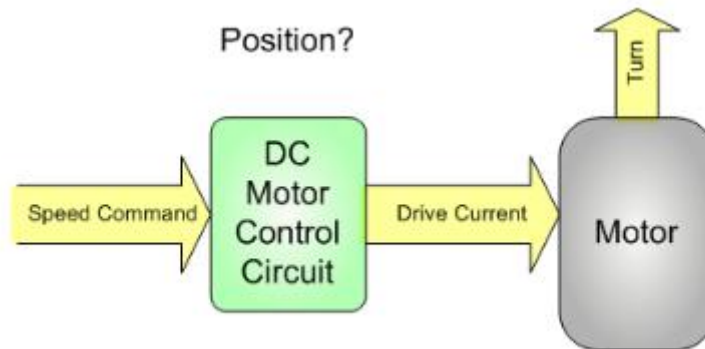
El control de creuer en un automòbil és un exemple d'un sistema servo. Mesura la seva velocitat i alimenta de nou a un circuit de control que ajusta l'accelerador per mantenir la velocitat.

Per la família Servomotor RC, la posició de l'eix de sortida es mesura i es reforma al circuit de control intern que s'ajusta el corrent al motor per mantenir la posició.



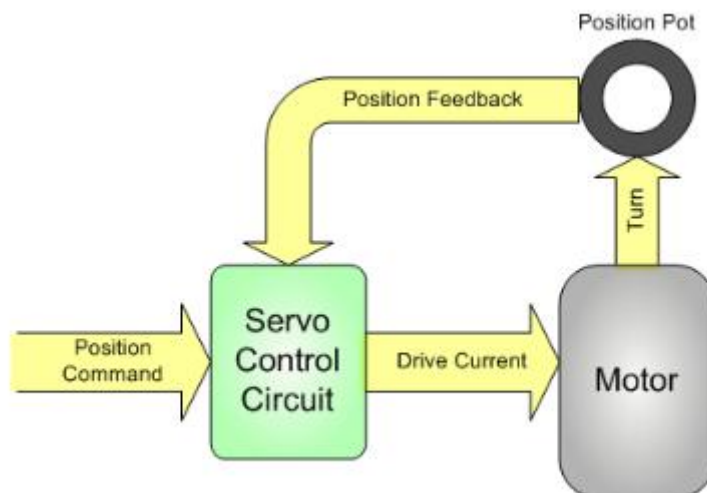
## Bucles oberts i tancats

Un sistema de "bucle obert" no té comentaris, de manera que no hi ha manera de verificar que estigui funcionant com s'espera. Una expressió comuna entre els enginyers de control és "No es pot controlar el que no es pot mesurar".



Il·lustració 21 Sistema bucle obert

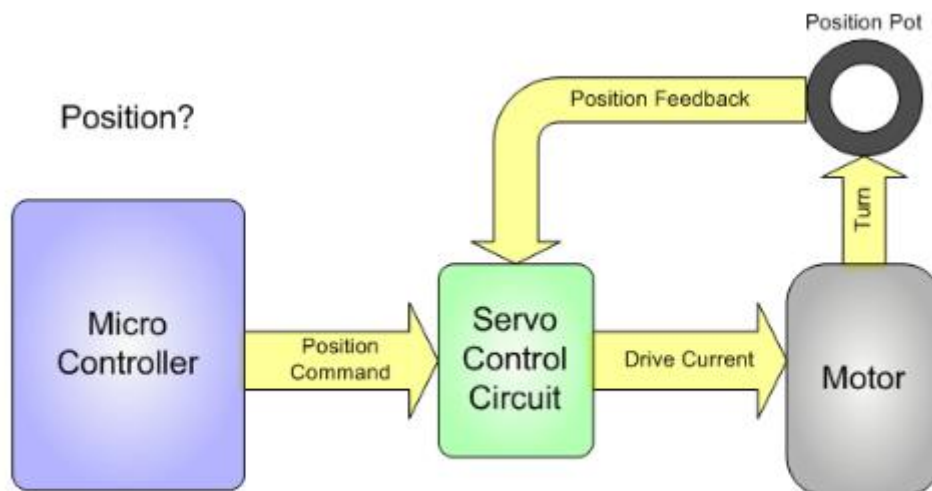
Un sistema de "Circuit Tancat" pot utilitzar el senyal de realimentació per ajustar la velocitat i la direcció del motor per aconseguir el resultat desitjat. En el cas d'un servomotor RC, la retroalimentació té la forma d'un potenciòmetre connectat a l'eix de sortida del motor. La sortida és proporcional a la posició de l'objectiu del servo



Il·lustració 22 Sistema de circuit tancat

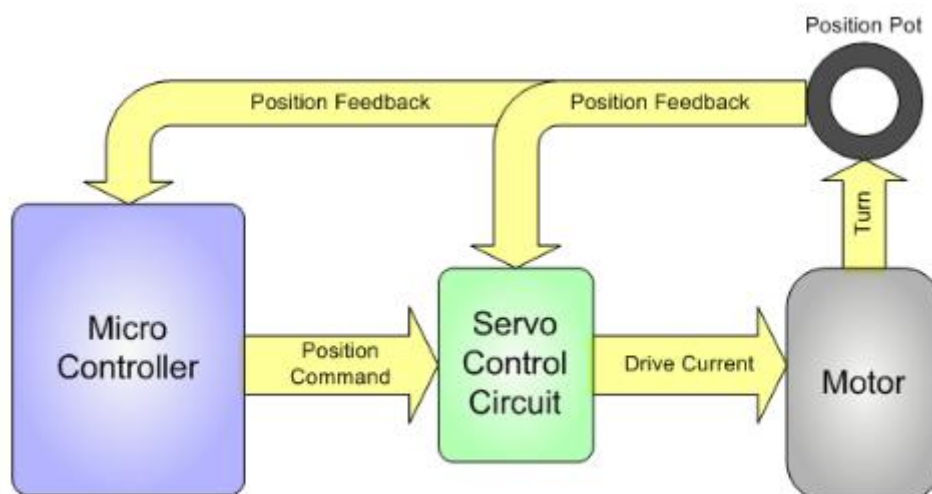
El problema amb el control d'un servomotor RC estàndard des d'un microcontrolador és que es tracta d'un "circuit tancat" dins de la caixa del servomotor, però de "circuit obert" respecte al microcontrolador. Pot dir-li al circuit de control de servo com vol que és col·loqui l'eix, però no té manera de confirmar si això succeeix realment.





II-lustració 23 Funcionament servomotor Standard

Els Servomotors Feedback li permet tancar aquest bucle extern en proporcionar el senyal de retroalimentació al microcontrolador.



II-lustració 24 Funcionament servomotor feedback

Els servomotor feedback que canviarem pels servomotor convencionals que porta el Robot braccio són de la marca Analog Feedback Servo i Analog Feedback Micro servo de la casa adafruit.

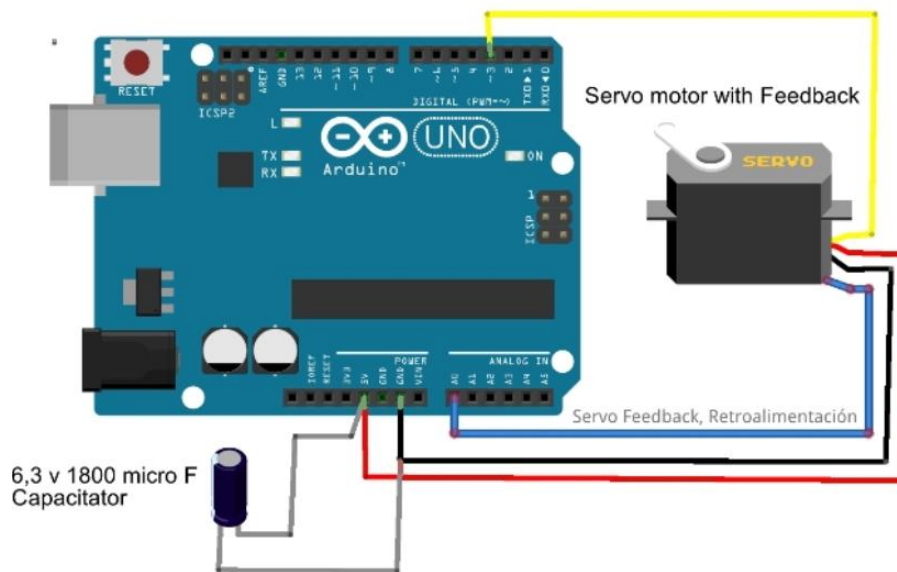
Aquest servomotors, per la realització d'aquest projecte eren idònies, perquè les característiques físiques, eren iguals que els servomotors que hi havien de canviar, ja que les mides havien de ser igual, perquè sinó, no podien fer el muntatge. La part de qualitat/preu dels productes tant en les especificacions dels productes que ens dona el fabricant, i els preus dels productes, eren perfectes pel desenvolupament del nostre projecte. Ven buscar diferents opcions de servomotors amb feedback d'altres fabricants, però no trobaven les mides correctes del servomotor o les especificacions dels productes eren iguals o més baixes, i els preus eren superiors als servomotors que havien elegit

En la taula 6 es pot observar les característiques del servomotor feedback

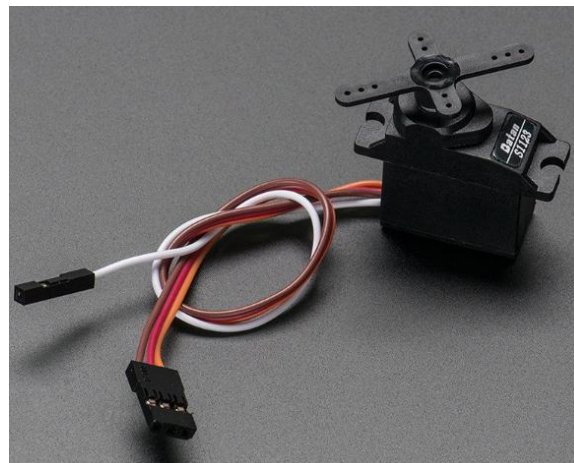
**Taula 5 Característiques de servomotor feedback**

<b>Analog Feedback Servo</b>		<b>Analog Feedback Micro servo</b>	
<b>Senyal de control</b>	PWM Analògica	<b>Senyal de control</b>	PWM Analògica
<b>Esforç de torsió</b>	6,5kg/cm	<b>Esforç de torsió</b>	1,8 kg/cm
<b>Velocitat</b>	0,21 sec/60º	<b>Velocitat</b>	0,1 sec/60º
<b>Rang de rotació</b>	180º	<b>Rang de rotació</b>	180º
<b>Dimensions</b>	39,55x39,55x19,5 mm	<b>Dimensions</b>	11,58x22,75x29,19 mm
<b>Pes</b>	46g	<b>pes</b>	15,81g

## Servo with Feedback, Retroalimentación



Il·lustració 25 Connexió servomotor feedback



Il·lustració 26 Analog Feedback Servo



Il·lustració 27 Analog Feedback Micro servo

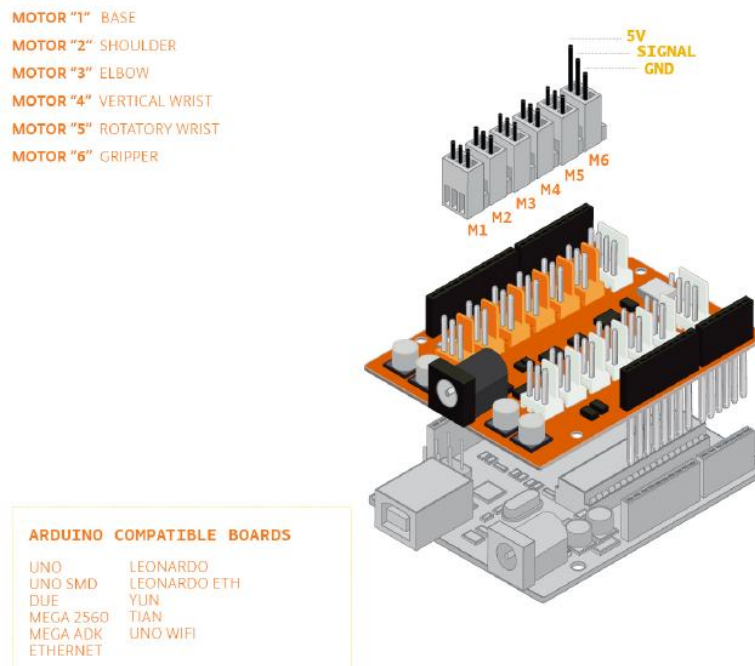
### 3.1.3 Shield Robot braccio

Si s'observa els pins de la Shield Proporcionat al kit, es pot comprovar que encaixa amb els pins de la placa D'Arduino UNO Rev3. Per aquesta raó, en la il·lustració 30 es mostra l'esquema intern de la Shield, proporcionat per Arduino.

Els pins de la placa de desenvolupament associat als motors M1,M2,M3,M4,M5 i M6 són els pins digitals 11,10,9,6,5 i 3 respectivament, ja que aquests pins digitals 1 i 2 són exclusivament de la comunicació sèrie amb la consola de l'IDE d'arduino.



Ilustración 28 Shield



**Il·lustració 29 Connexió shield amb l'arduino uno rev3**

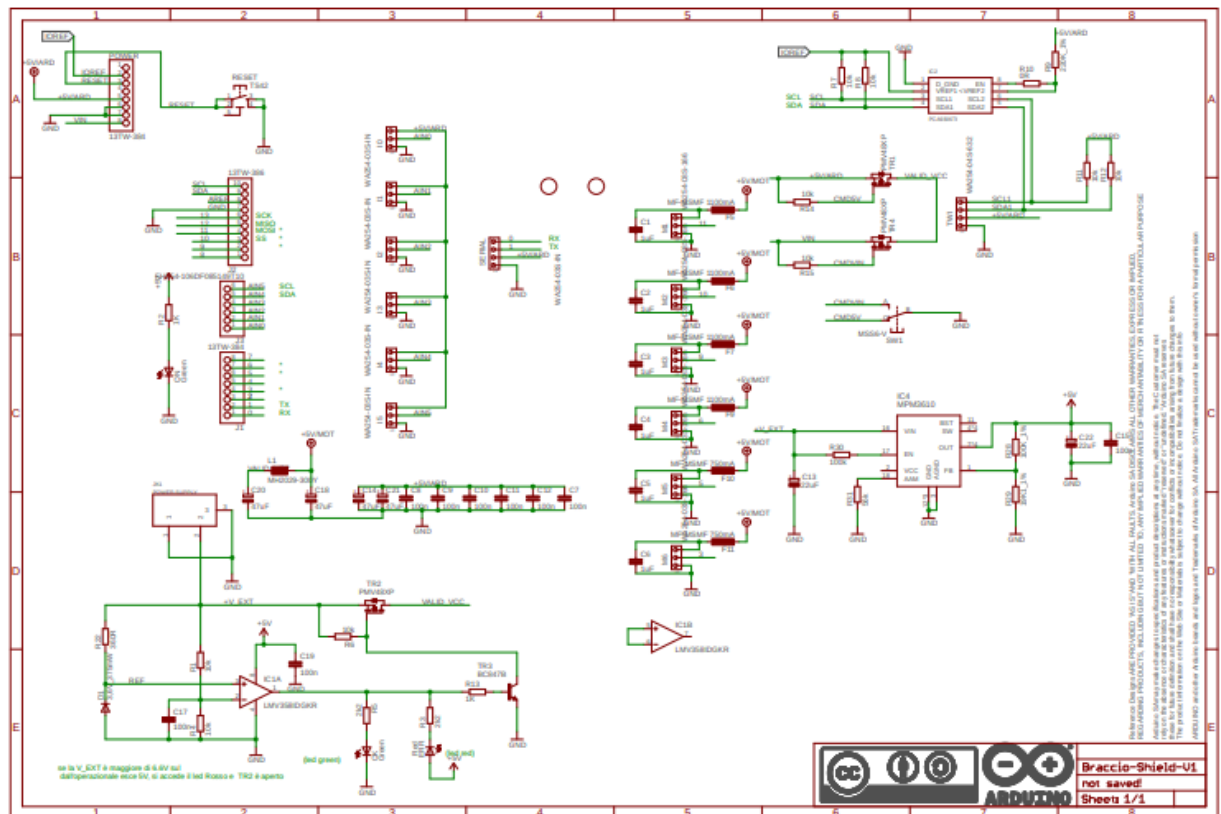


Ilustración 30 Esquema intern de la shield v1

## 4. Programació del robot braccio

Arduino proporciona un entorn de desenvolupament integrat (IDE) de codi obert, el llenguatge és una variant de programació en C.

Es un entorn senzill i potent per programar, però a més inclou les eines necessàries per compilar el programa i "cremar" el programa ja compilat en la memòria flash del microcontrolador. A més l'IDE ens ofereix un sistema de gestió de llibreries i plaques molt pràctic. Com IDE és un entorn de programació senzill que no té funcions avançades típiques d'altres IDEs, però suficient per programar.

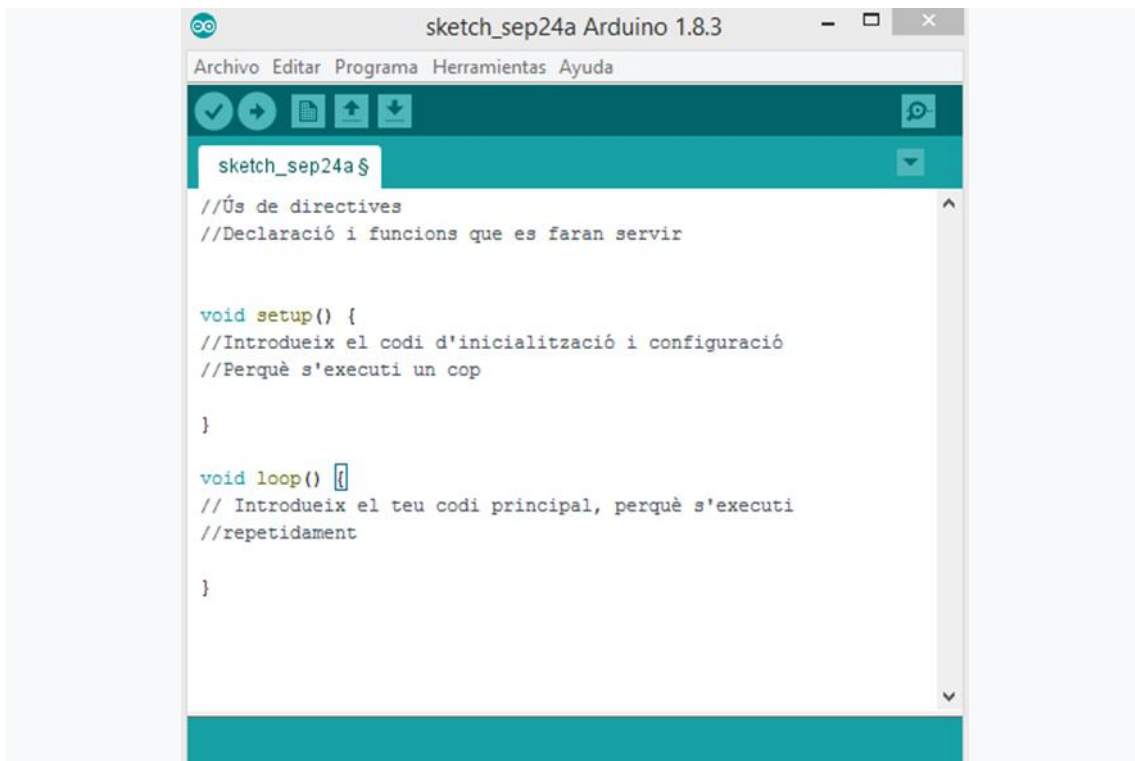
### 4.1 Estructura d'un sketch

Un programa d'Arduino s'anomena sketch o projecte i té l'extensió .ino

L'estructura bàsica d'un sketch d'Arduino és bastant simple i es compon d' almenys dues parts. Aquestes dues parts són obligatoris i tanquen blocs que contenen declaracions, estaments o instruccions.

Declaració de variable i funcions

- void setup (): Aquesta funció s'executa un sol cop al principi del programa i és on hem d'inicialitzar les variables creades anteriorment. Hem de tenir en compte que qualsevol variable creada dins de la funció, serà local, Això vol dir que la resta de programa no podran accedir-hi.
- void loop (): Es tracta d'un bucle infinit, en el qual hem d'implementar el codi que descriu el comportament que tindrà el microcontrolador, tenint en compte que aquest es repetirà de forma infinita.

**Il·lustració 31 Implementació de les funcions en Arduino IDE**

L'entorn de l'IDE es compon de menús desplegable (a dalt a l'esquerra) de la il·lustració 10, una petita consola que mostra els missatges relatius a la compilació i execució del codi (baix), el quadre de text en el qual s'introdueix el codi a executar i 6 botons amb les funcions de recopilar i executar el codi creat, obrir una nova finestra, obrir un arxiu, desar un arxiu modificat, obrir la consola de comunicació amb el microcontrolador, i obrir un desplegable amb els arxius dels quals es compon el projecte, respectivament i en ordre d'esquerra a dreta.

L'ús de l'entorn d'Arduino és molt estès per la facilitat que ofereix a l'hora d'implementar programes bàsics de control de maquinaria, ja que proporciona un gran nombre de llibreries que, en ser de codi obert, es van creant i millorant contínuament, gràcies a la gran comunitat que s'ha aconseguit. L'objectiu d'això és que qualsevol persona pugui dur a terme un projecte d'electrònica sense tenir a penes de coneixement sobre això. La contrapartida que presenta és que si es pretén objectius molt tècnics dins el projecte, l'ús d'aquestes llibreries deixa de ser recomanables per la quantitat de recursos que fa servir, tenint en compte que les característiques tècniques de les plaques d'Arduino són limitades.



## 4.2 Desenvolupament de l'algoritme

La primera cosa que hem de poder fer amb l'algoritme que es desenvoluparà, és moure el braç articulat Braccio, cosa que farem mitjançant l'ús de la llibreria "Servo.h" de codi obert que ens proporciona la web d'Arduino. Per a la programació d'aquests tipus de servomotors. En general, l'estructura d'aquests exemples implica de la creació d'un objecte global tipus braccio, l'inicialitzarem des d'un "setup" i, si es desitja un sol cicle de moviment, farem servir la funció de moviment del braç dins d'aquesta mateixa funció. D'una altra forma, s'haurà d'implementar dins de la funció "loop".

L'algoritme que desenvoluparem consta de les següents parts:

- **Declaracions de pins**
- **Calibració dels valors inicials**
- **Inicialització del servo**
- **Bucle de funcionament**

### 4.2.1 Declaració de pins

A la inicialització dels servomotors es declara els pins de control, que corresponen als pins digitals que designarem a cada un dels servomotors que anem a utilitzar, igualment hem de designar cada servomotor els pins analògics per poder utilitzar la realimentació, per poder calcular l'angle i poder visualitzar al monitor sèrie.

### 4.2.2 Calibratge de valors

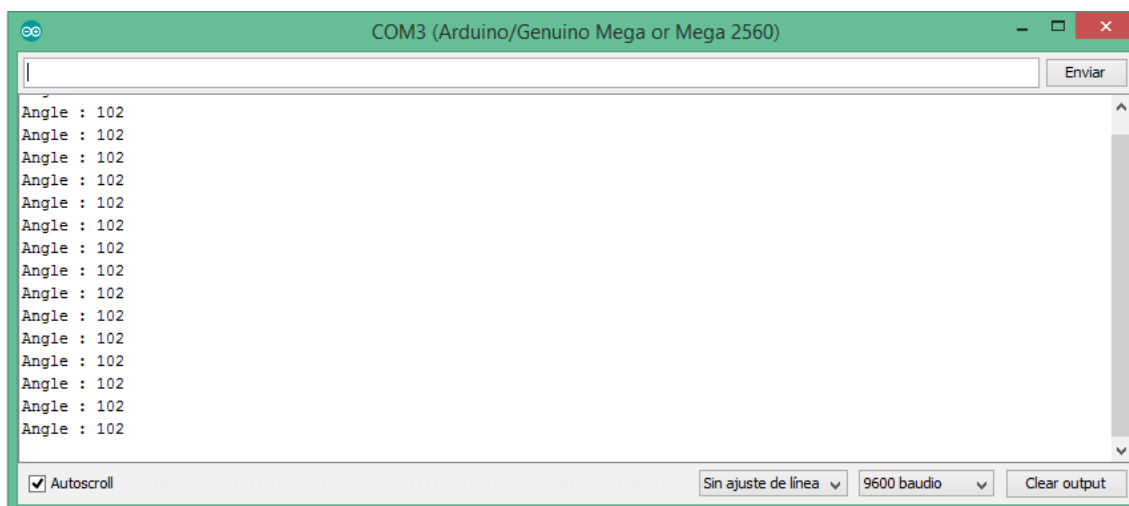
Per poder calibrar els valors dels angles del servomotor necessitem designar valors mínims i màxims que podrà anar el servomotor tant en els valors del control (pins digitals) com en els valors de retroalimentació (pins analògics), després declarem els mínims i els màxims dels servomotors tant en el de control i la retroalimentació. Començarem a calibrar el servomotor. Utilitzarem una tolerància de calibratge, en el nostre cas és de 2 perquè el fabricant del servomotor que utilitzarem ja ens ho han indicat en el manual d'utilització. Per calibrar el servomotor utilitzarem l'opció "Calibrate" on utilitzarem els valors mínims i màxims que hem designat anteriorment per poder iniciar el calibratge, on el servomotor es mourà a la posició màxima i mínima i després registrar el valor obtingut a la retroalimentació donant un retard en cada una d'aquestes etapes amb la funció "delay"

### 4.2.3 Inicialització del servo

Per inicialitzar el servomotor entrarem en la funció "setup" on utilitzarem l'opció "myservo" per poder inicialitzar el servomotor, també inicialitzem el monitor sèrie i finalment inicialitzem el calibratge dels servomotors, en aquest apartat només s'inicialitzarà cadascun dels elements que necessitem per poder moure i visualitzar els angles del servomotor.

### 4.2.4 Bucle de funcionament

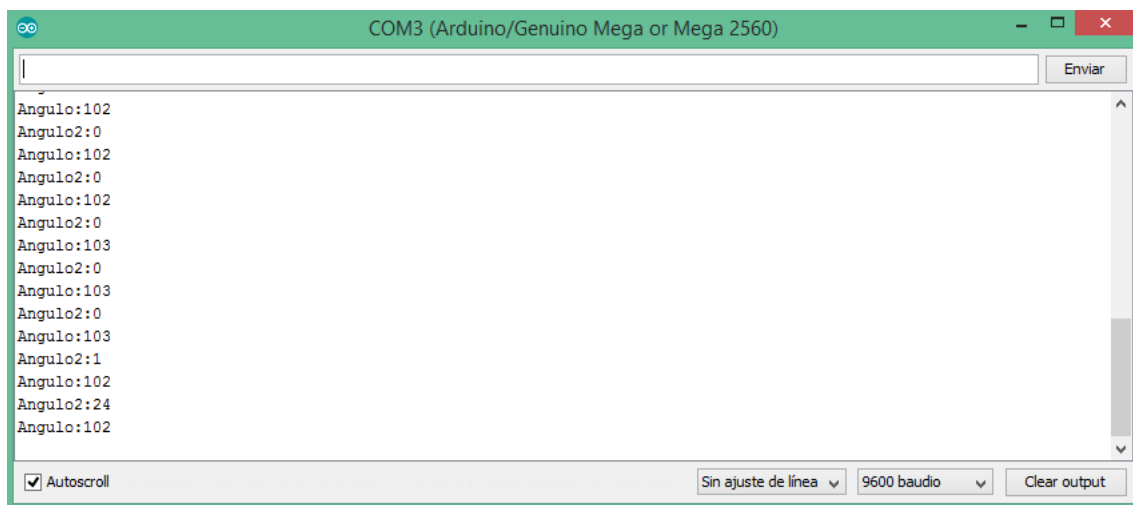
En l'apartat de funcionament entrarem en la funció loop (bucle) on es programarà els moviments del servomotor i la visualització del monitor sèrie per poder programar cada moviment. Utilitzarem l'opció "myservo.write ()" on escriurem els valors d'angles que volem utilitzar per fer el moviment del robot, amb l'opció "Serial.print (getPos ())" podrem visualitzar al monitor sèrie l'angle de retroalimentació de cada servomotor i finalment l'opció "abs (map ())" calcularem el valor de retroalimentació per a la posició final.



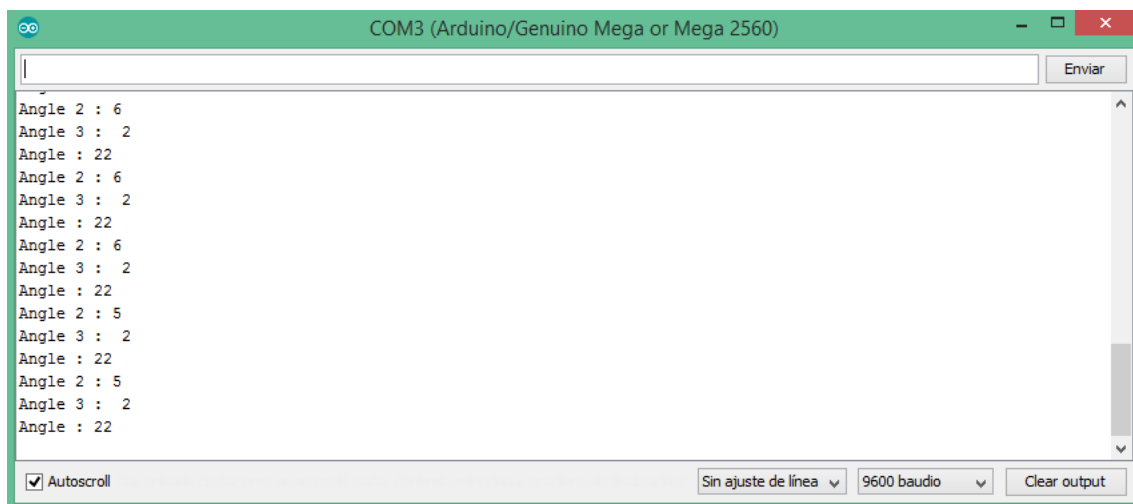
**Il·lustració 32 Visualització d'angles al monitor sèrie amb un servomotor amb feedback**

En la il·lustració 29 observen la visualització inicial amb un servomotor amb feedback.

Seguidament en les il·lustracions 30 i 31 observarem la visualització d'angles amb més d'un servomotor amb feedback.



**Il·lustració 33 Visualització d'angles al monitor sèrie amb mes d'un servomotor amb feedback**



**Il·lustració 34 Visualització angles al monitor amb mes d'un servomotor amb feedback**

## 5. Conclusions

La realització d'aquest treball ha suposat un augment considerable dels coneixements en la programació de l'autor, ja que ha hagut d'aprendre l'ús del llenguatge de programació d'Arduino i saber utilitzar bé l'entorn del programa.

Durant el procés de desenvolupament del projecte, es van presentar diversos inconvenients, alguns dels quals es presentaran a continuació. La resolució dels problemes ha suposat una oportunitat més per aprendre més coses

En primer lloc, hi ha errors en la visualització dels valors en el monitor sèrie. El senyal del feedback del servomotor és analògic, havia de tenir cura en la conversió analògica-digital dels valors obtinguts, atès que si aquesta conversió fallés, els valors que visualitzaríem al monitor sèrie serien errònies i una altra part important és en l'àmbit de programació, si aquest resultat no es guardava a la variable assignada correctament, això també influiria que els valors siguin incorrectes.

En segon lloc, cada vegada que s'incorporava un servomotor més, els valors que sortia al monitor sèrie del programa d'Arduino eren erronis. Per solucionar aquest problema havíem de modificar en la línia de codi de la part de calibratge, perquè cada servomotor inicialitzi aquesta funció individualment. Perquè si només s'inicialitzava una vegada, només calibrava el primer.

Com es pot observar, malgrat tots els inconvenients sorgits, l'objectiu del treball ha sortit satisfactòriament, tant per complir amb les especificacions plantejades com tot el que ha aportat el seu desenvolupament .

## 5. Conclusions

La realització d'aquest treball ha suposat un augment considerable dels coneixements en la programació de l'autor, ja que ha hagut d'aprendre l'ús del llenguatge de programació d'Arduino i saber utilitzar bé l'entorn del programa.

Durant el procés de desenvolupament del projecte, es van presentar diversos inconvenients, alguns dels quals es presenta a continuació.

En primer lloc, hi ha errors en la visualització dels valors en el monitor sèrie. El senyal del feedback del servomotor es analògic, havia de tenir cura en la conversió analògica-digital dels valors obtinguts, atès que si aquesta conversió fallés, els valors que visualitzaríem al monitor sèrie serien errònies.

En segon lloc, cada vegada que s'incorporava un servomotor més, els valors que sortia al monitor sèrie del programa d'Arduino eren erronis. Podria ser una mala programació del programa, que el programa calibrés un dels servos i la resta no el calibrava, per, això, hi havia errors de valors d' angles dels servomotors.

per aquests problemes sorgits en el desenvolupament del projecte, no he aconseguit l'objectiu de visualitzar correctament els angles dels servomotors.

## 6. Estudi Econòmic

Els punts clau d'aquest treball són que està realitzat de manera que no s'hagin de posseir coneixement tècnics previs. Això contribueix al fet que els alumnes de la universitat puguin utilitzar el robot i puguin desenvolupar projectes amb més varietats de tasques a resoldre i poder així entendre millor la robòtica

A continuació es presentaran el cost desglossat que ha suposat el desenvolupament d'aquest treball.

**Taula 6 Estudi econòmic**

Producte	Quantitat	Preu(€)	Total(€)
Ordinador portàtil	1	350	350
Arduino IDE	1	0	0
Servomotor feedback	4	15	15
Servomotor feedback	2	12	12
TinkerkitBraccio Robot	1	200	200
Arduino UNO Rev 3	1	20	20
Total			654

## 7. Bibliografía

TinkerkitBraccio Robot [En línia] disponible en :

<https://store.arduino.cc/tinkerkit-braccio> [consulta setembre 2019]

Parts d'Arduinounorev [En línia] disponible en :

<https://aprendiendoarduino.wordpress.com/tag/esquemas-electricos/>

[consulta setembre 2019]

Robòtica Industrial:una oferta nada sencilla [En línia] disponible en:

<https://walterfarah.wordpress.com/2013/06/04/robotica-industrial-una-oferta-nada-sencilla/> [consulta setembre 2019]

What is a Social robot? [En línia] disponible en:

<https://techforaging.com/social-robots> [consulta setembre 2019]

Classification of service robot by applications areas [En línia] disponible en:

[https://ifr.org/img/office/Service\\_Robots\\_2016\\_Chapter\\_1\\_2.pdf](https://ifr.org/img/office/Service_Robots_2016_Chapter_1_2.pdf)

[consulta setembre 2019]

About servos and feedback [En línia] disponible en :

<https://learn.adafruit.com/analog-feedback-servos/about-servos-and-feedback>

[consulta setembre 2019]

Using Feedback [En línia] disponible en :

<https://learn.adafruit.com/analog-feedback-servos/using-feedback>

Una nueva era en la robótica industrial [En línia] disponible en :

[https://www.aer-automation.com/wp-content/uploads/2018/05/Presentaci%C3%B3n\\_AER\\_jornada\\_Vigo.pdf](https://www.aer-automation.com/wp-content/uploads/2018/05/Presentaci%C3%B3n_AER_jornada_Vigo.pdf)

[consulta setembre 2019]

Robótica [En línia] disponible en :

<https://es.wikipedia.org/wiki/Rob%C3%B3tica> [consulta setembre 2019]

Leonardo Torres Quevedo [En línia] disponible en :

[https://es.wikipedia.org/wiki/Leonardo\\_Torres\\_Quevedo](https://es.wikipedia.org/wiki/Leonardo_Torres_Quevedo) [consulta setembre 2019]

Isaac Asimov [En línia] disponible en :

[https://es.wikipedia.org/wiki/Isaac\\_Asimov](https://es.wikipedia.org/wiki/Isaac_Asimov) [consulta setembre 2019]

## 8. Annex A: Codi implementat En Arduino

```
#include<Servo.h>

#include<SPI.h>

#include<Wire.h>

Servo myservo;

//Control and feedback pins

int servoPin=3;

int feedbackPin=A0;

Servo myservo1;

int servoPin1=5;

int feedbackPin1=A1;

Servo myservo2;

int servoPin2=6;

int feedbackPin2=A2;

Servo myservo3;

int servoPin3=9;

int feedbackPin3=A3;

Servo myservo4;

int servoPin4=10;

int feedbackPin4=A4;

Servo myservo5;

int servoPin5=11;

int feedbackPin5=A5;


//Calibration values

int minDegrees;

int maxDegrees;

int minFeedback;
```

```
int maxFeedback;

int tolerance=2;//max feedback measurement

void calibrate(Servo servo,int analogPin,int minPos, int maxPos)
{

    // Moure a la Posició Mínima i reemplaçar el valor de retroalimentació
    servo.write(minPos);

    minDegrees=minPos;

    delay(2000);

    minFeedback=analogRead(analogPin);

    // moure a la posició màxima i registrar el valor de realimentació
    servo.write(maxPos);

    maxDegrees=maxPos;

    delay(2000);

    maxFeedback=analogRead(analogPin);
}

void setup()
{

    myservo.attach(servoPin);
    myservo1.attach(servoPin1);
    myservo2.attach(servoPin2);
    myservo3.attach(servoPin3);
    myservo4.attach(servoPin4);
    myservo5.attach(servoPin5);

    Serial.begin(9600);

    calibrate(myservo,feedbackPin,0,160);//Calibrar para el rango de 0-160 grados
    calibrate(myservo1,feedbackPin1,0,160);//Calibrar para el rango de 0-160 grados
```



```
calibrate(myservo2,feedbackPin2,0,160);//Calibrar para el rango de 0-160 grados
calibrate(myservo3,feedbackPin3,0,160);//Calibrar para el rango de 0-160 grados
calibrate(myservo4,feedbackPin4,0,160);//Calibrar para el rango de 0-160 grados
calibrate(myservo5,feedbackPin4,0,160);//Calibrar para el rango de 0-160 grados
```

```
}
```

```
void loop()
```

```
{
```

```
myservo.write(100);
```

```
delay(2000);
```

```
Serial.print("Angle :");
```

```
Serial.println(getPos(feedbackPin1));
```

```
myservo1.write(90);
```

```
delay(2000);
```

```
Serial.print("Angle 1 :");
```

```
Serial.println(getPos(feedbackPin));
```

```
myservo2.write(70);
```

```
delay(2000);
```

```
Serial.print("Angle 2 :");
```

```
Serial.println(getPos(feedbackPin2));
```

```
myservo3.write(60);
```

```
delay(2000);
```

```
Serial.print("Angle 3 : ");
```

```
Serial.println(getPos(feedbackPin3));
```

```
myservo4.write(40);  
  
delay(2000);  
  
Serial.print("Angle 4 :");  
  
Serial.println(getPos(feedbackPin4));  
  
  
myservo5.write(50);  
  
delay(2000);  
  
Serial.print("Angle 5 :");  
  
Serial.println(getPos(feedbackPin5));  
  
  
}  
  
int getPos(int analogPin)  
{  
  
  Returnabs(map(analogRead(analogPin),minFeedback,maxFeedback,minDegrees,maxDegrees));  
  
}
```